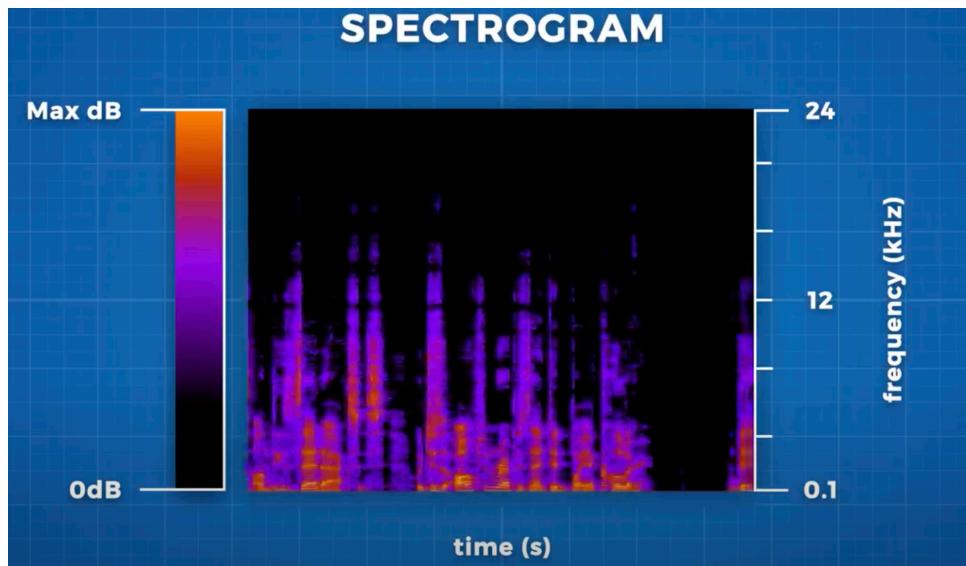




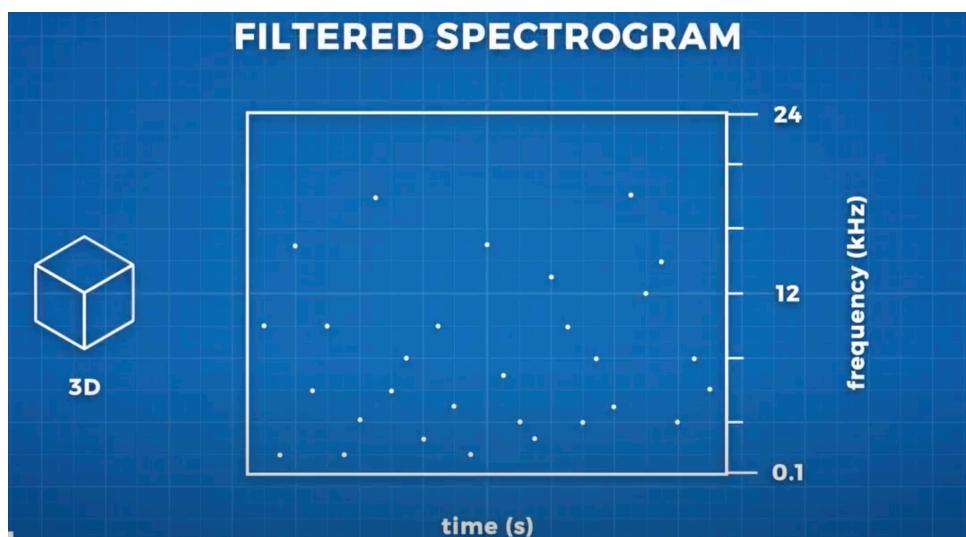
Shazam е мобилен софтуер, който ще назове всяка песен за секунди. Той ще идентифицира всяка музика, която звучи около вас, дори и звука да е „замърсен“ от други странични шумове от околната среда. Шазам открива съвпадение сред милиони песни в сървърната си база от данни. За повечето това може би ще им се струва като тривиална задача. Човешкият мозък може да идентифицира песен невероятно бързо от много ранни години, но пътищата в мозъка ни, които ни позволяват това да се случи са неуписуемо бързи и комплексни. Комбинация от специфични тонове наредени в специфичен ред ни позволява на мозъка да разпознаем песен от хилядите други които сме чуvalи. Но запитвали ли сте се някога как е възможно да програмирате компютъра да прави това вместо вас? Компютъра няма интуитивно разбиране за музиката. Той може единствено да сравнява една песен с друга песен от базата си данни. Но това е проблем, който може да се сравни с намирането на игла в купа сено като гледате например снимка на иглата и я сравнявате с всяка сламка сено - по цвят, дължина и т.н. докато накрая не я открием. Но когато мозъка ни чуе някакъв откъс от песен, той не проверява във всички песни които е чул до сега, за да търси сред тях, а специфични неврони се активират и го насочват къде точно да търси. По същия начин инженерите от Shazam са стигнали до извода, че може да разделят базата си данни на различни парчета, като например по музикални инструменти включени в тях. Всяка нота произведена от определен музикален инструмент, макар и една и съща нота, има различна честота на звучене. Всеки инструмент има специфична еволюция на базовия тон и оттенъчните му тонове, които се създава от определена нота. Още в самото начало става ясно, че в софтуерната структурата на шазам, огромно значение заема представянето на базата данни, тя е разделена на много парчета и подслоеве, на които ще обърнем внимание по-долу.

За човешкия мозък е много лесно да разграничи звук от пиано и звук от китара, но на компютъра му е необходим предварително зададен начин как се изразяват количествено тези характеристики, за да може да ги разпознае. Тук се намесва спектрограмата. Спектрограмата е визуална презентация на звуковите вълни. Това е 3D графика с време по абсцисата, честота по ординатата и амплитуда (или с други думи - силата) по

апликатата. Като за опростяване, апликатата се изразява графично с цвят, вместо реална 3D графика.



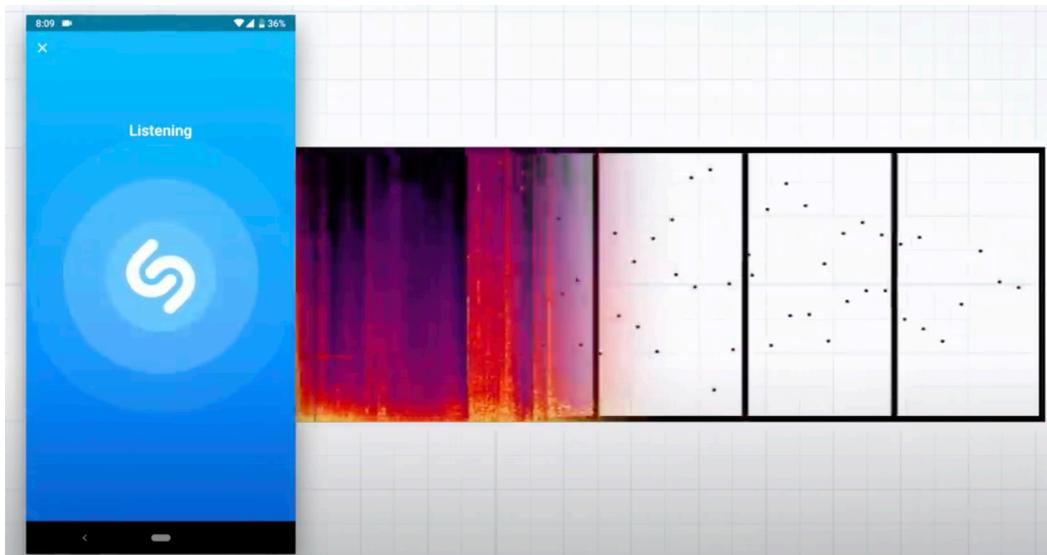
Тази 3D графика е нещо което компютъра с лекота може да разпознае и съхрани в база данни, НО в спектрограмата има огромно количество данни и колкото повече данни има - толкова повече време за пресмятания е необходимо за да се намери съвпадение. Първата стъпка за намаляне на времето за пресмятанията е като се намят параметрите необходими за разпознаването на песен. За целта Shazam използва така наречения *fingerprint*. Точно така, това е същия този пръстов отпечатък, за който учихме в курса по СДА и не е нищо повече от едно малко-по сложно хеширане. Те трансформират тези спектрографи в нещо което изглежда като карта със звездички по нея.



Тук всяка звездичка (или в случая точица) представлява най-високата честота за определено време. Правейки това, Shazam не само редуцира графиката от 3 деменции на 2, но и драстично намалява количеството данни на графиката. Това е първата съществена стъпка в технологията на Shazam. Всяка една песен в базата данни на Shazam е съхранена чрез нейн пръстов отпечатък. Когато натиснете бутона на приложението за разпознаване на песен, Shazam достъпва микрофона ви и създава пръстов отпечатък на звуковите вълни, които чува.

Shazam е направен така, че създава пръстови отпечатъци и данни само на звуковете, които се отклояват най-силно. След като веднъж приложението е създало този отпечатък на аудиото, той се изпраща на сървърите на Shazam, където е сърцето на софтуера. Там започва частта по разпознаването. Тук нещата стават много сложни и специфични. Нека разгледаме опростен отпечатък от някоя песен и записан отпечатък на същата песен в базата данни на Shazam.

Записания отпечатък е само кратък откъс от песента (3 - 5 секунди). В нашия пример имаме само 3 записани честоти (приемаме, че всеки записан отпечатък има 3 възможни записани честоти). Ако искаме да проверим за съвпадение какво ще направим? Естествено ще обходим линейно и ако срещнем съвпадение ще проверим и останалите честоти. Тоест ще имаме 10 проверки умножени по 3 и умножени по всяка една песен в шазам, което както се досещате е твърде бавно. Това не е начина, по който Shazam търси за съвпадение.



Първо Shazam категоризира отпечатъка по много хитър начин. Shazam не търси дали дадена нота се съдържа в дадена песен, а търси дали няколко ноти разделени от определен интервал от време се съдържат, също както го прави човешкия мозък. Това се превръща в адрес за търсене в хеш таблицата. Хеширането и Хеш таблиците са мощена техника и инструментариум, който намира приложение не само тук, ами и почти навсякъде където има търсене в огромно количество данни.

Не би било редно да се говори за Shazam и да не се споменат редовете на Фурие.

Pulse Coded Modulation (PCM) е импулсно кодирана модулация която представлява цифрови сигнали. Използва се от компактни дискове и повечето електронни устройства. Това накратко е поток от организирани битове. Този модул си го има вграден вътре в самата слушалка на устройството. Преминаването на аналогов звук към дигитален е вече лесна задача, но Shazam работи само с честоти, а как да ги получим тях? Тази част е много важна тъй като алгоритъма за пръстови отпечатъци на песни работи само с тях (с честотите). **Тук се намества ДИС-а. За аналогови (и следователно непрекъснати) сигнали има трансформация, наречена непрекъсната трансформация на Фурие. Това е функция която превръща функцията на времето във функция на честотите.** (Жозеф Фурие е френски математик, роден през далечната 1768 г. в Оксер, който доказва че всяка една графика на функция, може да се представи като безкрайна сума от синуси и косинуси - т.н. ред на Фурие, който се използва в Shazam. С други думи, ако приложим трансформацията на Фурие върху звук - тя ще ни даде честотите и техните интензитети вътре в този звук. Дали Фурие е подозирал, че труда му ще струва над 400 милиона долара след около 250 години?). Просто казано - в сървъра на Shazam е имплементиран алгоритъм, който прилага трансформацията на дискретната функция на Фурие върху моно песни. Тоест преди това всяка стерео песен се трансформира в моно. Става на въпрос за песента, която се складира в базата данни. Макар, че не е толкова просто, ще спрем да дълбаем в тази част до тук, въпреки че е доста интересна.

Модул на сървъра:

Shazam предварително изчислява пръстови отпечатъци от много голяма база данни с музикални песни. Всички тези пръстови отпечатъци се поставят в база данни с отпечатъци, която се

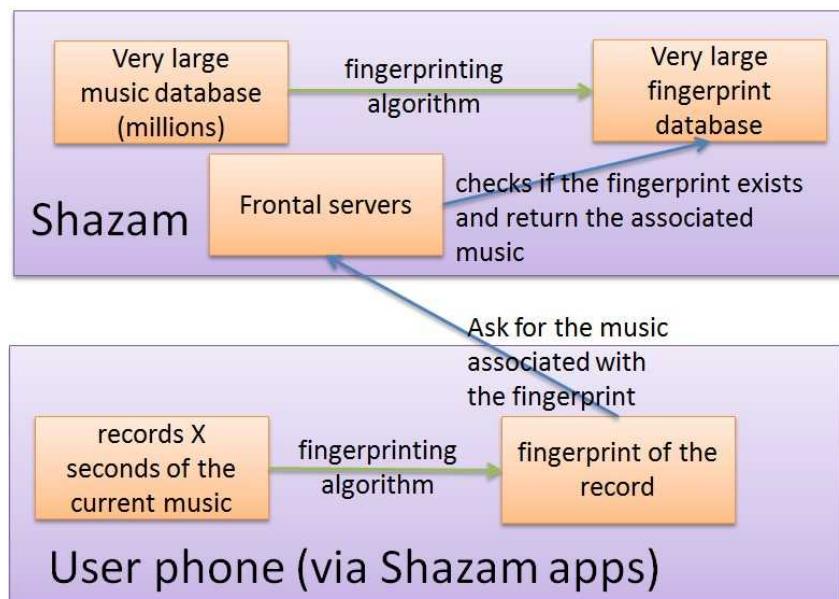
актуализира всеки път когато се добави нова песен в базата данни с песни (Тази база данни е силно разделена на главни и подразделения, за по лесно търсене).

Модул на клиента:

Когато потребителят използва приложението Shazam, приложението първо записва текущата музика с микрофона на телефона, телефонът прилага същия алгоритъм за пръстови отпечатъци като Shazam. Shazam проверява дали точи пръстов отпечатък съвпада с един от пръстовите му отпечатъци. Ако отговора е „да“, той връща на потребителя мета данни свързани с песента (име на песента, URL адрес, и т.н.).

В приложението на Shazam има модул който се грижи за това, микрофона да е толерантен към звуковите грешки (това са тези странични звуци, за които споменахме в началото).

Като цяло е много трудно да се намери информация относно структурата на процесите на това интересно приложение, но дори и да има налична такава дадена от Shazam, то този процес, който е в сървъра и се грижи за създаването на цялостен отпечатък на всяка нова песен като я трансформира от стерео в моно и след това чрез редове на Фурье изкарва честотните данни от нея би бил иключително интересен да се види, макар и трудно разбираем.



Източници:

■ <http://coding-geek.com/how-shazam-works/>

<https://medium.com/@treycoopermusic/how-shazam-works-d97135fb4582>

<https://ehikioya.com/audio-fingerprinting-how-shazam-works/>